## Гайд для роботи в команді на GitHub

### 1. \*\*Налаштування репозиторію\*\*

- Один із учасників створює новий репозиторій на GitHub і запрошує інших як колабораторів.

- Кожен розробник клонує репозиторій на свій локальний комп’ютер:

git clone https://github.com/username/repository.gi

### 2. \*\*Основні команди Git\*\*

#### \*\*2.1. Перевірити статус репозиторію\*\*

Команда показує поточний стан репозиторію та які файли змінилися:

git status

```

#### \*\*2.2. Додати зміни до індексу\*\*

Щоб додати файл для коміту, скористайтеся:

git add <назва\_файлу> # Або git add . для всіх змін

```

#### \*\*2.3. Створити коміт\*\*

Закріпіть зміни комітом з описом:

git commit -m "Опис змін"

```

#### \*\*2.4. Оновити зміни з віддаленого репозиторію\*\*

Перед тим як пушити зміни, отримайте останні зміни з віддаленого репозиторію:

git pull origin main

```

#### \*\*2.5. Запушити зміни на GitHub\*\*

Відправте зміни на віддалений репозиторій:

git push origin <назва\_гілки>

```

### 3. \*\*Алгоритм роботи над новими фічами або задачами\*\*

1. \*\*Перевірка основної гілки та оновлення\*\*:

- Переключіться на основну гілку (`main`):

git checkout main

git pull origin main

```

2. \*\*Створення нової гілки\*\*:

- Створіть нову гілку для своєї задачі:

git checkout -b feature/new-feature

```

3. \*\*Зміни у новій гілці\*\*:

- Вносьте зміни у код, додавайте файли, комітьте:

git add .

git commit -m "Опис змін"

```

4. \*\*Завантаження змін з основної гілки перед пушем\*\*:

- Переконайтесь, що в основній гілці немає нових змін:

git checkout main

git pull origin main

git checkout feature/new-feature

git merge main

```

5. \*\*Запушити зміни в нову гілку\*\*:

- Пуште зміни на GitHub:

git push origin feature/new-feature

```

6. \*\*Створити Pull Request (PR)\*\*:

- Після завершення роботи створіть PR для злиття ваших змін у основну гілку на GitHub.

### 4. \*\*Як завантажити віддалену гілку, якщо її немає локально\*\*

Якщо у віддаленому репозиторії є гілка (наприклад, `user`), але на локальному репозиторії вона відсутня, дійте так:

1. \*\*Оновити інформацію про віддалені гілки\*\*:

git fetch

```

2. \*\*Перевірити наявність гілки\*\*:

Переконайтесь, що гілка є у списку віддалених:

git branch -r

```

3. \*\*Створити локальну копію віддаленої гілки\*\*:

git checkout -b user origin/user

```

4. \*\*Оновити гілку перед початком роботи\*\*:

Після того як ви отримали гілку, зробіть її синхронізацію:

git pull origin user

```

### 5. \*\*Злиття та вирішення конфліктів\*\*

1. \*\*Оновити основну гілку перед злиттям\*\*:

Переконайтесь, що ви працюєте з найсвіжішою версією основної гілки:

git pull origin main

```

2. \*\*Злиття гілки\*\*:

Зливати гілку після вирішення всіх коментарів і перевірок у PR.

3. \*\*Вирішення конфліктів\*\*:

Якщо при злитті виникають конфлікти, виправте їх вручну, додайте файл до індексу, та створіть коміт:

git add <файл>

git commit -m "Вирішено конфлікт"

```

### Додаткові рекомендації:

- \*\*Часті коміти\*\*: Це допоможе легко відслідковувати історію змін.

- \*\*Регулярна синхронізація з основною гілкою\*\*: Робіть `git pull` перед початком роботи, щоб уникати конфліктів.

- \*\*Ясні коментарі до комітів\*\*: Використовуйте зрозумілі та короткі повідомлення при комітах.